

CSCI 1312 (Introduction to Programming for Engineering), Fall 2017

Homework 2

Credit: 30 points.

1 Reading

Be sure you have read (or at least skimmed) the assigned readings from chapters 2 and 3.

2 Honor Code Statement

Please include with each part of the assignment the Honor Code pledge or just the word “pledged”, plus one or more of the following about collaboration and help (as many as apply).¹ Text *in italics* is explanatory or something for you to fill in. For written assignments, it should go right after your name and the assignment number; for programming assignments, it should go in comments at the start of your program(s).

- This assignment is entirely my own work. (*Here, “entirely my own work” means that it’s your own work except for anything you got from the assignment itself — some programming assignments include “starter code”, for example — or from the course Web site. In particular, for programming assignments you can copy freely from anything on the “sample programs page”.*)
- I worked with *names of other students* on this assignment.
- I got help with this assignment from *source of help — ACM tutoring, another student in the course, the instructor, etc.* (*Here, “help” means significant help, beyond a little assistance with tools or compiler errors.*)
- I got help from *outside source — a book other than the textbook (give title and author), a Web site (give its URL), etc..* (*Here too, you only need to mention significant help — you don’t need to tell me that you looked up an error message on the Web, but if you found an algorithm or a code sketch, tell me about that.*)
- I provided help to *names of students* on this assignment. (*And here too, you only need to tell me about significant help.*)

3 Programming Problems

Do the following programming problems. You will end up with at least one code file per problem. Submit your program source (and any other needed files) by sending mail to `bmassing@cs.trinity.edu` with each file as an attachment. Please use a subject line that mentions the course and the assignment (e.g., “csci 1312 hw 2” or “CS1 hw 2”). You can develop your programs on

¹Credit where credit is due: I based the wording of this list on a posting to a SIGCSE mailing list. SIGCSE is the ACM’s Special Interest Group on CS Education.

any system that provides the needed functionality, but I will test them on one of the department's Linux machines, so you should probably make sure they work in that environment before turning them in.

1. (15 points) Write a C program to convert a Fahrenheit temperature to Celsius. The rule for converting Fahrenheit temperature F to Celsius temperature C is

$$C = (5/9)(F - 32)$$

The program should ask the user for the Fahrenheit temperature and print the equivalent Celsius temperature. You can use integers or floating-point numbers for this problem.

For this assignment only, you do not need to do any kind of checking that what the user enters is actually numeric (integer or floating-point, depending on your code), since we haven't yet talked about conditional execution. Just assume it is and do the required calculations.

2. (15 points) Write a C program that prompts the user for a number of seconds S (S should be a non-negative integer, but your program does not need to check for that) and prints S in a more normal notation — years, days, hours, minutes, and seconds. Assume there are 365 days in a year (not exactly right but makes the calculations simpler). For example, 100 seconds is 1 minute and 40 seconds. (You may be interested to run this program using successive powers of 10 as input and see how long it takes before seconds start to add up to years.)

For this assignment only, you do not need to do any kind of checking that what the user enters is actually an integer and non-negative, since we haven't yet talked about conditional execution. Just assume it is and do the required calculations.